

FILE EXTENSION AUTO REPAIR DENGAN TEKNIK DETEKSI HEADER

Agung Budi Prasetyo

STMIK AKAKOM Yogyakarta
Jl. Raya Janti 143, Karangjambe Yogyakarta
081 5688 0231
e-mail : agung_bp@akakom.ac.id

Abstrak

File Extension merupakan sebuah penanda untuk menunjukkan ciri dari sebuah file seperti file gambar dapat diketahui karena memiliki extension bertipe gambar seperti .GIF, .BMP, .JPG, dsb, file video dapat diketahui karena memiliki extension bertipe video seperti .AVI, .MPG, .DAT, dsb. Pada implementasi yang ada file extension juga menjadi dasar dari sebuah operasi pencarian data oleh search engine. Namun demikian karena suatu sebab, terkadang file extension dapat hilang misalnya karena virus atau karena perbedaan versi software pembuatnya maupun sistem operasinya.

Permasalahan seperti di atas dapat diatasi dengan teknik deteksi header dimana ciri dari sebuah file tidak semata didasari oleh extensionnya, namun oleh informasi yang tersimpan pada header file itu sendiri. Dengan menggunakan bahasa delphi, operasi-operasi file dengan teknik deteksi header bisa dilakukan.

File extension auto repair dengan teknik deteksi header dirancang dan diciptakan dengan tujuan memperbaiki extension dari file dengan cara mengenali sebuah file, baik yang tidak memiliki extension maupun file yang extensionnya telah dimanipulasi, kemudian memperbaikinya dan mengembalikan file tersebut ke bentuk extensionnya yang semula.

Kata Kunci : Deteksi Header, File Extension, Repair

Pendahuluan

File adalah entitas dari data yang disimpan di dalam sistem berkas yang dapat diakses dan diatur oleh pengguna. Sebuah file memiliki nama file dan extension yang unik dalam direktori di mana ia berada.

Nama file diberikan terhadap sebuah file agar file dapat dikelola dengan mudah. Meski oleh sistem operasi penamaan file dilakukan dengan menggunakan angka-angka biner, sistem operasi akan menerjemahkan angka-angka biner tersebut menjadi karakter yang mudah dipahami.

Extension merupakan sebuah penanda dari sebuah file yang digunakan untuk menunjukkan ciri-ciri yang memungkinkan kita untuk mengetahui secara spesifik termasuk jenis apakah file yang kita simpan dalam sebuah penyimpanan elektronik. Misalnya file gambar dapat diketahui karena memiliki extension bertipe gambar seperti .GIF, .BMP, .JPG, dsb, ataupun file video dapat diketahui

karena memiliki extension bertipe video seperti .AVI, .MPG, .DAT, dsb. Di sisi lain extension sering juga menjadi dasar dari sebuah operasi pencarian file oleh search engine.

Namun demikian mungkin anda pernah mengalami keadaan dimana sebuah file hilang ataupun rusak extensionnya karena suatu hal, misalnya karena terserang virus, karena terjadi perubahan versi software yang membuat file tersebut, maupun karena perubahan versi dari sistem operasi yang digunakan. Disebut hilang karena pada file tidak ada lagi extensionnya, dan disebut rusak karena file extensionnya tidak hilang tetapi mengalami perubahan menjadi extension lain tidak sesuai dengan tipe file yang sesungguhnya.

Karena keadaan di atas, maka file extension tidak dapat lagi menjadi penanda tipe file yang valid bagi file yang bersangkutan. Ekstensi dari keadaan ini adalah *associate* dari file tersebut dalam file manager (misalnya windows explorer) menjadi kacau, bahkan juga dapat menyebabkan tidak

terjaringnya file yang bersangkutan oleh mesin pencari yang mendasarkan metode pencariannya dengan file extension.

Dengan menggunakan teknik deteksi header, telah berhasil dibangun sebuah sistem berbasis windows, yang mampu melakukan identifikasi dan rekonstruksi terhadap sebuah file yang mengalami kerusakan maupun kehilangan extensionnya. Teknik ini mampu mengenali sebuah file yang bermasalah dengan cara mengenali headernya, menganalisanya dengan cara mencocokkan dengan database yang dimilikinya, dan kemudian file tersebut dikembalikan pada extensionnya yang semula.

Teknik deteksi header ini pernah diterapkan pada sebuah mesin pencari file yang mencari file pada sebuah PC, dimana pencarian file tersebut tidak dilakukan berdasarkan extension file, namun berdasarkan pembacaan dan penganalisaan header dari sebuah file. Oleh mesin pencari ini mampu ditemukan file-file, bahkan terhadap file-file yang tidak dilengkapi dengan type file (extension), ataupun file-file yang extensionnya sengaja disamarkan atau diganti dengan extension yang bukan miliknya. Pengujian terhadap software tersebut telah memberikan hasil berupa kesuksesannya dalam menemukan file-file yang tidak mampu ditemukan dengan mesin pencari lain

2. Dasar Teori

Secara umum, format file dibedakan dalam 2 jenis yang berbeda yaitu File ASCII dan File Non ASCII.

File ASCII adalah sebuah file yang hanya mengandung teks-teks yang diformat dengan menggunakan pengkodean ASCII. File ini hanya terdiri atas karakter, angka, tanda baca, tabulasi, dan karakter pemisah baris (carriage return). Dalam file ASCII, tidak ada pemformatan yang ekstensif seperti dalam dokumen Microsoft Word atau Rich Text Format, melainkan pemformatan yang digunakan hanyalah pemformatan standar yakni ASCII.

Dalam sistem operasi Windows, file ASCII ini dapat dibuat dengan menggunakan program Notepad atau editor teks lainnya. Jenis extension yang digunakannya adalah .txt (default), .bat, .cmd, .ini, .inf, dan masih banyak lainnya. Sistem operasi lainnya juga mengimplementasikannya, tapi beberapa sistem operasi (utamanya adalah keluarga UNIX) tidak mengharuskan adanya extension file seperti pada Windows. Normalnya, sebagian besar file dalam sistem operasi UNIX merupakan file ASCII, kecuali file yang dapat dieksekusi (program). File ASCII umumnya digunakan sebagai

file teks biasa, skrip, kode sumber sebuah program (source code sebuah bahasa pemrograman), file konfigurasi (terutama dalam keluarga sistem operasi UNIX), atau file teks lainnya. Bahkan, banyak juga orang yang membuat gambar dengan menggunakan file ASCII, yang sering disebut sebagai ASCII Art. Karena file ASCII mengandung teks yang tidak diformat, file tersebut dapat dipahami banyak sistem operasi, karena memang ASCII adalah sebuah standar internasional (Windows, Macintosh, dan UNIX mendukung sepenuhnya standar ini). Akibatnya, file tersebut dapat menjadi sangat berguna dalam rangka berbagi informasi antar platform atau aplikasi.

File Non ASCII adalah file-file dengan pengaturan atau pemformatan khusus yang tidak bisa dibaca langsung dengan teks editor. File-file non ASCII ini antara lain: File Gambar, File Video, File Dokumen, dan lain-lain.

File Gambar adalah file komputer yang digunakan untuk menyimpan citra. Contoh extension file gambar antara lain : BMP, JPG, JPEG, JPE, JFIF, GIF, TIF, TIFF, PNG, dan lain-lain

File video adalah file komputer yang digunakan untuk menyimpan kumpulan berkas digital seperti video, audio, metadata, informasi, pembagian chapter, dan judul sekaligus, yang dapat dimainkan atau digunakan melalui perangkat lunak tertentu pada komputer. Contoh extension file video antara lain : WAV, AIFF, IFF, MOV (untuk program Quicktime), MPEG-1, MPEG-2, MPEG-4 atau MP4, Ogg, ASF (standard file video dari Microsoft selain WMV), RM (untuk program Real Video), MKV (Matroska), MKA (Matroska), 3gp (untuk telepon genggam), NUT, MPEG, MXF, ratDVD, SVI, DivX, DAT (file standard vcd), VOB (file standard dvd), AVI (standard file video dikomputer), VCD, QT (file video program Quicktime), m1v (file MPEG video), m2p, m2v, MPG, WMV, ASX, SWF (file animasi macromedia), AMV (digunakan pada perangkat MP4 player), dan lain-lain.

File Dokumen adalah file-file yang diciptakan dengan menggunakan software-software pembuat dokumen seperti Word Processing, Spreadsheet, Database, macam-macam editor (editor gambar, editor music, editor video) dan software-software aplikasi lain. Contoh extension file dokumen antara lain : DOC, XLS, ODT, DBF, PDF, CDR, dan lain-lain.

Teknik Deteksi header adalah sebuah teknik yang digunakan untuk mengenali sebuah file dengan membuka file tersebut kemudian membaca

informasi berupa kode-kode yang terdapat pada bagian awal dari sebuah file. Dengan informasi tersebut, sebuah file dapat diketahui identitasnya tanpa perlu melihat extensionnya

Dukungan Bahasa Delphi

Bahasa pemrograman Delphi menyediakan instruksi-instruksi dan fungsi-fungsi yang dapat digunakan untuk hal-hal yang berkaitan dengan pengoperasian file.

FindFirst, FindNext, dan FindClose

Digunakan untuk men-scanning file-file yang ada dalam sebuah directory / folder dengan memberikan set attribute dan direktori yang sesuai.

Unit : SysUtils
Category : file management routines
Delphi syntax : function FindFirst(const Path: string; Attr: Integer; var F: TSearchRec): Integer;
Description : FindFirst mencari file di dalam folder yang ditunjuk oleh Path dan mencocokkannya dengan atribut yang ditunjuk oleh parameter Attr. Hasil pencarian dapat dilihat dalam parameter F. FindFirst akan menghasilkan nilai returns 0 jika file berhasil ditemukan, jika tidak berhasil akan muncul error code.
 Parameter Path berisi directory atau folder dimana file yang akan dicari berada, misalnya 'C:\test*.*'.
 Parameter Attr adalah spesifikasi dari file yang hendak di-scan meliputi :

Constant	Description
faReadOnly	Read-only files
faHidden	Hidden files
faSysFile	System files
faVolumeID	Volume ID files
faDirectory	Directory files
faArchive	Archive files
faAnyFile	Any file

ExtractFileName(sr.Name)

Digunakan untuk men-scanning file-file yang ada dalam sebuah directory / folder dengan memberikan set attribute dan direktori yang sesuai.

Unit : SysUtils
Category : file name utilities

Delphi syntax : function ExtractFileName(const FileName: string): string;
Description : Fungsi ini akan menghasilkan nama file dan extensionnya.

ExtractFileExt(sr.Name)

Digunakan untuk men-scanning file-file yang ada dalam sebuah directory / folder dengan memberikan set attribute dan direktori yang sesuai.

Unit : SysUtils
Category : file name utilities
Delphi syntax : function ExtractFileExt(const FileName: string): string;
Description : Fungsi ini akan menghasilkan extension file dari variabel FileName.

FileOpen, FileSeek, dan FileRead

Digunakan untuk membuka file yang telah ditentukan nama filenya.

Unit : SysUtils
Category : file management routines
Delphi syntax : function FileOpen(const FileName: string; Mode: LongWord): Integer;
Description : Perintah FileOpen digunakan untuk membuka file yang telah dideklarasikan dalam file handle. Nilai acces mode ditentukan dari nilai fmOpen yang didefinisikan dengan perintah fileOpen sebelumnya. Jika nilai balik yang didapat dari fungsi ini bernilai 0 atau lebih besar maka penggunaan fungsi ini berhasil, namun jika bernilai -1 maka artinya telah terjadi error.
 Biasanya operasi ini digunakan bersama pemakaian AssignFile, Rewrite, and Reset.

AssignFile, Reset, Rewrite, CloseFile

Unit : System
Category : file management routines
Delphi syntax : procedure AssignFile(var F; FileName: string);
Description : Dalam penggunaan AssignFile, diperlukan dua buah parameter yaitu parameter FileName yang digunakan untuk menentukan file apa yang hendak di'buka', dan parameter F yang digunakan untuk menampung isi file yang telah dibuka oleh perintah AssignFile.

Setelah file berhasil dibuka, ada 2 jenis mode operasi yang dapat dilakukan yaitu *Reset(F)* digunakan untuk membaca isi file, serta *Rewrite(F)* yang digunakan jika hendak memanipulasi ini dari file.

Operasi yang telah dibuka dengan *AssignFile* harus diakhiri dengan perintah *CloseFile(F)*.

RenameFile

Digunakan untuk mengganti nama file.

Unit : SysUtils
Category : file management routines
Delphi syntax :

```
function RenameFile(const
OldName, NewName: string):
Boolean;
```


Description : *RenameFile* akan berusaha mengubah nama file dari nama file lama (*OldFile*) menjadi nama file baru (*NewFile*). Jika operasi berhasil, fungsi *RenameFile* akan menghasilkan nilai *returns true*. Sedangkan jika fungsi *RenameFile* tidak berhasil mengubah nama file (misalnya jika aplikasi tidak memberikan hak akses (permission) untuk memodifikasi file, maka fungsi *RenameFile* akan menghasilkan nilai *returns false*.

Rename

Digunakan untuk mengganti nama file.

Unit : System
Category : IO routines
Delphi syntax :

```
procedure Rename(var F;
Newname: string);
```


Description : File eksternal yang diasosiasikan dengan *F* di-rename menjadi *Newname*. *F* adalah variabel yang dapat dideklarasikan sebagai string maupun char.

ChangeFileExt

Digunakan untuk mengganti nama file extension.

Unit : SysUtils
Category : file name utilities
Delphi syntax :

```
function ChangeFileExt(const
FileName, Extension: string):
string;
```

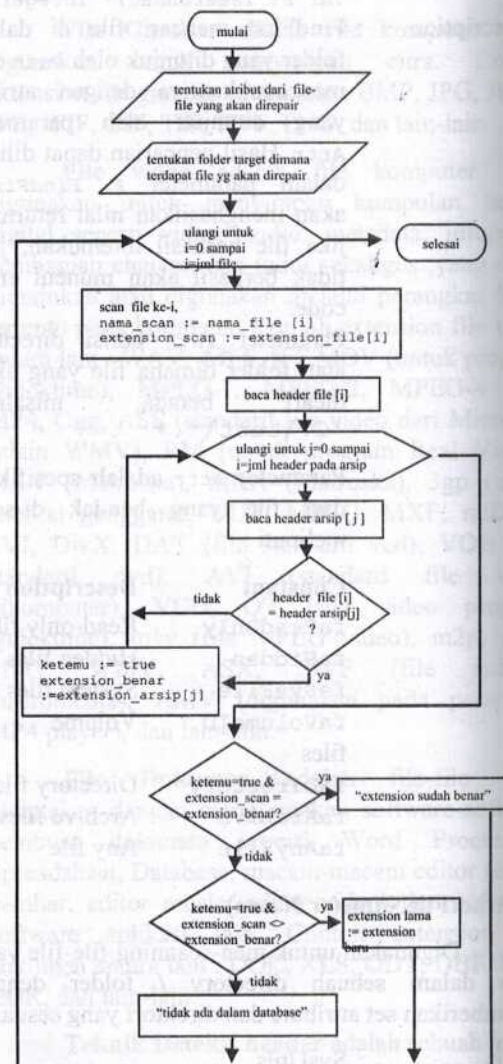

Description : *ChangeFileExt* akan mengambil nama file dari konstanta *FileName* dan mengubah extension dari file

dengan extension yang ada dalam konstanta *Extension*. *Extension* menetapkan extension baru, termasuk inisial dot character. Fungsi ini tidak merubah nama file, tetapi akan membuat file baru dengan nama extension yang baru.

Aplikasi File Extension Auto Repair

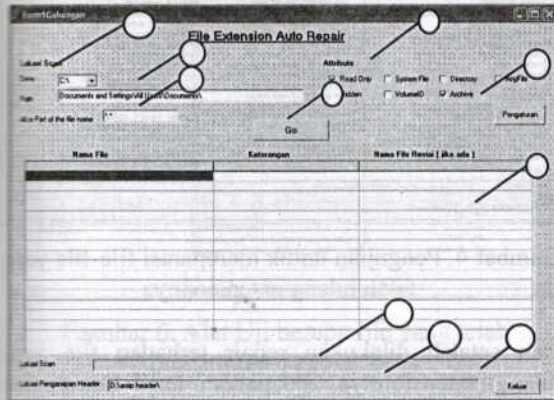
Berdasarkan kemampuan pemrograman Delphi dalam mengakomodir pengelolaan file dengan deteksi header di atas kemudian diciptakanlah sebuah program aplikasi untuk mereparasi file-file yang telah hilang extensionnya, termasuk juga memperbaiki file-file yang extensionnya telah dimanipulasi dengan extension yang tidak sesuai.

File Extension Auto Repair dibangun dengan flowchart sebagaimana disajikan pada gambar 1.



Gambar 1. Flowchart File Extension Auto Repair

Dengan dukungan bahasa delphi, serta pemodelan dengan flowchart di atas, dibangunlah sebuah interface untuk File Extension Auto Repair ini dengan format form sebagaimana tersaji pada gambar 2.



Gambar 2. Model Perancangan Interface

Keterangan :

1. ComboBox1 : untuk menentukan drive lokasi pencarian
2. Edit1 : untuk menentukan path lokasi pencarian
3. Edit2 : untuk tentukan nama & tipe file yg dilacak.
4. CheckBox1 : untuk menentukan atribut apa saja yang akan disertakan dalam pelacakan.
5. Button3 : untuk setting header-header.
6. Button1 : untuk melakukan proses pencarian
7. StringGrid1 : untuk menampilkan hasil pencarian.
8. Edit3 : untuk melihat directory yang sedang discan
9. Edit4 : untuk beri informasi lokasi pengarsipan
10. Button2 : untuk mengakhiri program

Berdasarkan flowchart serta desain interface di atas, disusunlah script untuk file extension auto repair ini dengan susunan script sebagaimana disajikan pada cuplikan program di bawah ini

```
//MENGLIK TOMBOL CARI
procedure TForm1.Gabungan.Button1Click(Sender: TObject);
var
  sr, ssr
  TSearchRec;

  nama_file_dokumentasi, folder_dokumentasi : string;
  urutan : integer;

  header_dari_file_yg_discan, file_yg_discan : string;
  nama_file_yg_discan, ext_file_yg_discan : string;
  ext_file_yg_benar, BackupName : string;
  efyd : string[255];

  drive_scan, path_scan, all_scan : string;

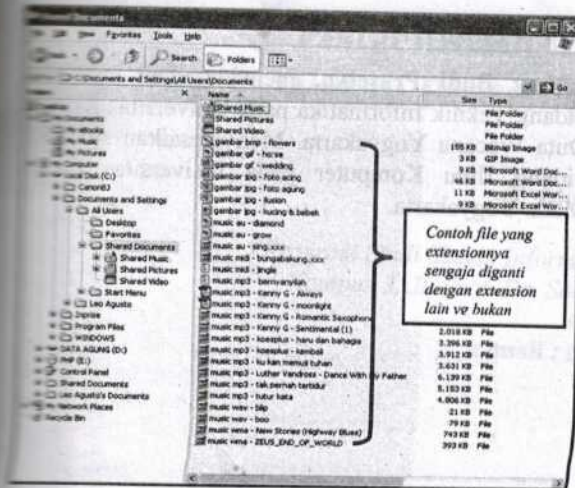
  iFileHandle, iFileLength, iBytesRead : Integer;
  Buffer : PChar;
  F1 : TextFile;
  Ch : String;
  ketemu : Boolean;
  FileAttrs, FfileAttrs, i : integer;
  f : file;
```

```
begin
  //CHECKBOX UTK TENTUKAN ATRIBUT FILE YANG DI SCAN
  StringGrid1.RowCount := 1;
  if CheckBox1.Checked then FileAttrs := faReadOnly
  else FileAttrs := 0;
  if CheckBox2.Checked then FileAttrs := FileAttrs
  +faHidden;
  if CheckBox3.Checked then FileAttrs := FileAttrs
  +faSysFile;
  if CheckBox4.Checked then FileAttrs := FileAttrs
  +faVolumeID;
  if CheckBox5.Checked then FileAttrs := FileAttrs
  +faDirectory;
  if CheckBox6.Checked then FileAttrs := FileAttrs
  +faArchive;
  if CheckBox7.Checked then FileAttrs := FileAttrs
  +faAnyFile;

  //MENENTUKAN LOKASI SCAN
  drive_scan := ComboBox1.Items.Strings
  [ComboBox1.ItemIndex];
  path_scan := Edit1.Text;
  all_scan := Edit2.Text;
  Edit3.Text := drive_scan + path_scan + all_scan;

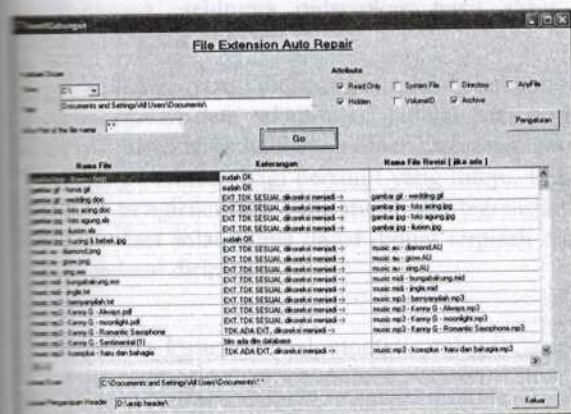
  with StringGrid1 do
  begin
    RowCount := 0;
    //mulai men-scan file dalam 1 folder -----
    if FindFirst(drive_scan + path_scan
  +all_scan, FileAttrs, sr)=0
  then begin
      repeat
        //scan file ke - i
        file_yg_discan := drive_scan + path_scan + sr.Name;
        //memperoleh nama & extension file yang di scan
        nama_file_yg_discan := sr.Name;
        ext_file_yg_discan := '';
        efyd := ExtractFileExt(sr.Name);
        for i := 2 to StrLen(PChar(string(efyd))) do
          ext_file_yg_discan := ext_file_yg_discan +
        efyd[i];
        if (sr.Attr and FileAttrs) = sr.Attr then
          begin
            try
              //buka header file ke- i -----
              iFileHandle := FileOpen(file_yg_discan,
            fmOpenRead);
              iFileLength := FileSeek(iFileHandle, 0, 2);
              FileSeek(iFileHandle, 0, 0);
              Buffer := PChar(AllocMem(iFileLength + 1));
              iBytesRead:= FileRead(iFileHandle, Buffer^,
            iFileLength);
              //headerdimasukkan
              kvariabel'header_dari_file_yg_discan'
            header_dari_file_yg_discan := Buffer;
              FileClose(iFileHandle);
              finally
                FreeMem(Buffer);
              end;
              //mulai men-scan file dokumentasi -----
              ketemu := false;
              folder_dokumentasi:=Edit4.Text;
              FfileAttrs := 0;
              FfileAttrs := FfileAttrs + faArchive;
              if FindFirst(folder_dokumentasi+'*.*',
            FileAttrs, ssr)=0
              then begin
                  repeat
                    //BUKA FILE DOKUMENTASI ke i
                    AssignFile(F1, folder_dokumentasi + ssr.Name);
                    Reset(F1);
                    while not Eof(F1) do
                      begin
                        Readln(F1, Ch);
                        //PROSES PENCOCOKAN DTBASE vs
                        HDR_DR_FILE_YG DISCAN
                        if header_dari_file_yg_discan = Ch then
                          begin
                            ketemu := true;
                            if ssr.Name[3]='.' then
                              ext_file_yg_benar:=ssr.Name[2];
                            if ssr.Name[4]='.' then
                              ext_file_yg_benar:=ssr.Name[2]+ssr.Name[3];
                            if ssr.Name[5]='.' then
                              ext_file_yg_benar:=ssr.Name[2]
                                +ssr.Name[3]+ssr.Name[4];
                            break;
                          end;
                        CloseFile(F1);
                        until FindNext(ssr) <> 0;
                        end; //end if (findfirst..)
                      //menampilkan pd stringGrid & merename jika
                      diperlukan -

```

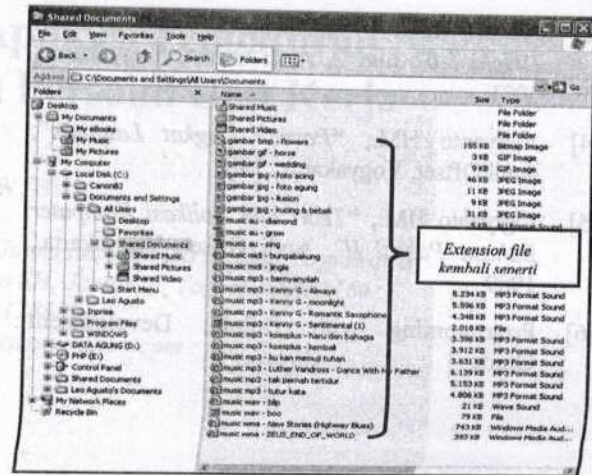
Gambar 6. Alat Uji berupa file yang telah dimanipulasi extensionnya

File-file yang telah dimanipulasi extensionnya sebagaimana gambar 6 kemudian diperbaiki dengan program File Extension Auto Repair sebagaimana disajikan pada gambar 7.



Gambar 7. Pengujian untuk mereparasi file-file yang telah dimanipulasi extensionnya

Dan setelah dilakukan perbaikan terhadap file yang telah dimanipulasi extensionnya tersebut, didapatkan hasil berupa file dengan extension yang valid sebagaimana disajikan pada gambar 8.



Gambar 8. Hasil menunjukkan file-file yang telah dimanipulasi extensionnya telah kembali normal

Dengan pembuktian-pembuktian di atas, dapat dilihat bahwa File Extension Auto Repair dengan Teknik Deteksi Header telah dapat memperbaiki file-file yang telah hilang extensionnya, termasuk juga memperbaiki file-file yang extensionnya telah dimanipulasi dengan extension yang tidak sesuai.

4. Kesimpulan

Dengan memanfaatkan teknik deteksi header, sebuah file dapat dikenali walaupun tidak ada lagi extensionnya.

Pengarsipan header file yang lengkap dapat memberi dukungan terhadap kinerja File Extension Auto Repair dengan Teknik Deteksi Header ini dalam mengenali sebuah file dan mengembalikan extension file ke dalam file dengan extensionnya yang semula.

File Extension Auto Repair dengan teknik deteksi header memungkinkan pemakai yang memiliki file yang telah hilang ataupun rusak extension, dapat memperbaiki extension file tersebut dengan mudah.

Daftar Pustaka

- [1] Agung Budi Prasetyo, "Search Engine Untuk Mencari File Dengan Deteksi Header", Format Volume 10 Nomor 3, September 2009.
- [2] Abdul Kadir, "Dasar-dasar Pemrograman Delphi 5.0" Jilid 1, Andi Offset, Yogyakarta., 2001

- [3] Abdul Kadir, "Dasar-dasar Pemrograman Delphi 5.0" Jilid 2, Andi Offset, Yogyakarta., 2001
- [4] Jogiyanto HM., "Pascal Tingkat Lanjutan", Andi Offset, Yogyakarta., 1988
- [5] Jogiyanto HM., "Teori dan Aplikasi Komputer Bahasa Pascal II", Andi Offset, Yogyakarta., 1989
- [6] Programming with Delphi, Development Guide

[CV Penulis]

Agung Budi Prasetyo, menyelesaikan studi S1 bidang Teknik Informatika pada Universitas Kristen Duta Wacana Yogyakarta. Menyelesaikan studi S2 bidang Ilmu Komputer pada Universitas Gadjah Mada Yogyakarta.

DAFTAR SUSUNAN PANITIA

PROGRAM COMMITTEE

Prof. H. Adhi Susanto, M.Sc, Ph.D. (UGM)
Prof. Dr. Ir. Achmad Djunaedi, MUP. (UGM)
Prof. Dr. Ir. Prayoto, M.Sc. (STMIK AKAKOM)
Prof. Dr. I Wayan Simri Wicaksana, S.Si, M.Eng. (Universitas Gunadarma)
Prof. Drs. Setiadji, S.U. (STMIK AKAKOM)
Prof. Drs. Suryo Guritno, M.Stat, Ph.D. (UGM)
Ir. P. Insap Santosa, M.Sc., Ph.D. (UGM)
Dr. Ir. Inggriani Liem (ITB)
Dr. Ir. Titon Dutono, M.Eng. (PENS-ITS/Dir. Telekomunikasi Ditjen Postel Menkominfo)
Dr. Ir. Sasongko Pramono Hadi, DEA.(Dir. ST Multimedia MMTC)
Ir. Lukito Edi Nugroho, M.Sc., Ph.D. (UGM)
Ir. Joko Lianto Buliali, M.Sc., Ph.D. (ITS)
Drs. Retyanto Wardoyo, M.Sc., Ph.D. (UGM)
Dr. LT Handoko (LIPI)

PELAKSANA SEMINAR

Pelindung

Ketua STMIK AKAKOM - Sigit Anggoro, S.T., M.T.

Penanggung Jawab

Kepala Puslitbang dan PPM - Dra. F. Wiwick Nurwiyati M.T.

Tim Pengarah :

Penibantu ketua 1 - Berta Bednar, S.Si. M.T.
Pembantu Ketua 2 - Ir.Totok Suprawoto, M.T., M.M.
Pembantu ketua 3 - Sri Redjeki, S.Si., M.Kom.

Ketua Panitia

L.N.Harnaningrum, S.Si., M.T.

Bendahara

Ir.Hera Wasiati (koord.)
Cicilia Sri Hastiwi

Sekretariat, Publikasi dan Kerjasama:

Agung Budi Prasetyo, S.Kom., M.Kom. (koord.)
Dison Librado, S.E., M.Kom.
Femi Dwi Astuti, S.Kom.
H. Sri Widodo
Endang Sekti W.
Nailus Sa'adah
Rita Darundia